Template for the Measurement Analysis & Reporting Procedure

Number: 580-TM-070-01
Effective Date: October 10, 2007
Expiration Date: October 10, 2012

Approved By: (signature)
Name: John Donohue
Title: Associate Division Chief, 580

Responsible Office: 580/Information Systems Division (ISD) **Asset Type:** Template
Title: Template for the Measurement Analysis and Reporting Procedure **PAL Number:** 3.4.1.3

Purpose	This template enables GSFC software projects to develop a Measurement Analysis and Reporting Procedure using a standard format. This procedure can stand as a separate document, or be added to a Software Management Plan / Product Plan (SMP/PP) as an appendix.
Scope	This template is recommended for all GSFC software projects and is required for projects producing Class A, B or C software, as defined in NPR 7150.2. (The NPR can be found at http://nodis.hq.nasa.gov/.) **Class A: Human-Rated Software Systems** **Class B: Non-Human Space Rated Software Systems** **Class C: Mission Support Software**
Roles and Responsibilities	**Product Development Lead (PDL)** • Responsible for the production of a Measurement Analysis and Reporting Procedure in conjunction with project planning
Tailoring of this Template	Text within the template that appears in this (style name = "Normal") style is equally applicable to every Measurement Data Collection and Storage Procedure (a.k.a. "the Procedure") and should be included without modification. All document section headings should also be included without modification, although their style names vary depending on outline level. *Text in this style [style name = "TAILORING ADVICE"] within the template is advice on how to tailor the information in any specific section. As the Procedure is developed, the generic [TAILORING ADVICE] text should be replaced with material that applies to the specific software project, or deleted, if it is general advice.* Text in **bold blue font** within brackets indicates places in the template where project-specific information needs to be inserted. For example, substitute the actual product name for the "**[Product Name]**" text in the Procedure title.

The gray-shaded paragraphs should remain in the tailored procedure as added guidance to team members.

Although the text in this template assumes the use of spreadsheet tools provided by the Software Process Improvement (SPI) project (see http://software.gsfc.nasa.gov), use of other tools is in no way precluded. The template is intended to document what a team is actually doing, so edit accordingly.

Note

The entire first two pages of this document should be deleted when producing the Procedure. This material is not part of the project's Procedure.

Change History

Version	Date	Description of Improvements
1.0	10/05/07	Initial version approved by CCB

[Product Name] Measurement Analysis and Reporting Procedure

Purpose	This procedure describes the measurement analysis and reporting procedures for the **[Product Name]** Product Development Team.
Scope	This procedure applies for the duration of the work performed by the Product Development Team. It is intended to support the team's periodic status monitoring and reporting as the project is progressing.
Roles and Responsibilities	**PDL/Team Lead** • Analyzes measures and records analyses. • Stores updated analysis spreadsheets back to the measurement repository • Reports results of analysis as part of status reviews **Guidance:** The mechanics of copying and storing analysis files may be delegated; the analysis and reporting itself may not.
Step-Action Table	This procedure is presented in three parts. The first is a step/action table that provides the sequence of steps to be followed to analyze and report a project's measures. The second part, presented in Table 1, is detailed information about each type of measurement, including the tools used to collect the measurement data, method and frequency of analysis, and the person responsible for analyzing and reporting that particular data. The information presented in this table informs every step of the procedure. The third part, presented below table 1, is a set of detailed instructions for analyzing each chart. These instructions inform each step of the procedure for a given type of data; steps 2 and 3 (analysis and documentation, respectively) form the heart of these instructions. NPR 7150.2 measurement areas provide the organization for both Table 1 and the instructions for each chart. The actual measures are defined in **[Table X of Section y]** in the Software Management Plan / Product Plan;

Step	Action	Role
1	Create a copy of each chart or table to be analyzed, using the instructions in the User's Guide. • For spreadsheets (whether SPI or equivalent tool), create a copy of the file containing charts using "save as" command. For data stored within a tool, generate the appropriate report. • Incorporate date into file name. **Guidance:** If using an SPI tool, the leftmost tab is a User's Guide documenting how this is done. If not using an SPI tool for a given chart, the project must write instructions for generating the chart.	**[PDL]**
2	Analyze the measurement data for each chart according to the instructions below. The measurement chart in the main body of the SMP/PP summarizes the analysis approach and defines the measures used. **Guidance:** The User's Guide in SPI tool also provides more detailed data definitions for that tool.	**[PDL]**
3	Document the results of measurement data analysis in the format of analysis/impact/corrective action. **Guidance:** The SPI measurement tools provide pre-packaged charts with blank analysis boxes. By convention these are the rightmost tabs at the bottom of the spreadsheet.	**[PDL]**
4	Transfer appropriate charts into the status review file as part of preparing for a BSR. **Guidance:** Choose charts relevant for the project's current life cycle phase.	**[PDL]**
5	Store files containing charts, tables and analysis results in the project measurement repository located at **[location]** (see DML). **Guidance:** This activity is in addition to the work being done to produce status report charts. Any analysis not presented in status reports should be recorded within the appropriate tool so that the analysis is documented and retained. **Guidance**: When using SPI tools, the charts, tables and analyses are generally in the new copy created in Step 1 of this procedure.	**[PDL/Team Lead]**

Table 1: Information for each type of measure collected

Tool	Chart/Table to Analyze	Analysis Schedule	Responsibility
Software Progress			
SPI Staffing Tool	18 Mo Chart_Proj	**[First of the month]**	**[PDL]**
	18 Mo Chart_Proj Cum	**[First of the month]**	
	Process Effort	**[First of the month]**	
SPI Schedule Tool	Detailed Schedule	**[First of the month]**	**[PDL]**
SPI Point Counting Tool	Trend Data	**[Every other Friday]**	**[DTL, TTL]**
Software Functionality			
SPI Reqts. Metrics Tool	Functionality by Build	**[First of the month]**	**[PDL]**
Software Requirements Volatility			
SPI Reqts. Metrics Tool	RQ Growth	**[First of the month]**	**[PDL]**
	RQ volatility	**[First of the month]**	
	RQ TBDs	**[First of the month]**	
Software Quality			
SPI Problem Report Tool	Status by Date	**[First of the month]**	**[PDL]**
	Status by CSCI	**[First of the month]**	
	Sev. By Date	**[Quarterly]**	
	Sev. By CSCI	**[Quarterly]**	
SPI Inspection Metrics Tool	Status Data	**First of the month**	**[PDL]**

Notes on Tailoring Table 1

- *Edit Table 1 to replace tool, chart or table names with those that the team is actually using. The names here are from the SPI tools.*
- *A product development team may collect and analyze multiple spreadsheets with the same type of measures if measurement and analysis for a subset of the software is delegated to a sub-team lead (Development Team Lead (DTL) or Test Team Lead (TTL)). In this case, each sub-team spreadsheet needs to documented in a separate row.*

Analysis Instructions

This set of analysis instructions defines how the 5 steps in the step/action table are carried out for each type of data. For the tables and charts generated by the SPI tools:

- Step 1 (creating charts) is carried out in accordance with the User's Guide for the relevant tool. In most cases this is either done automatically or through a few straightforward steps.
- Steps 2 and 3 (analysis of measures and documentation of results) are carried out according to the instructions below. The SPI tools provide blank boxes in which to write the analysis, impact, and corrective action.
- Step 4 (transferring appropriate charts to BSR) is done by copying the data to the BSR file by copying the chart and pasting it using the Paste Special | Picture (Enhanced metafile on PCs) command from the Edit menu.

For non-SPI tools, the instructions below also contain steps for creating charts and transferring charts to the BSR.

In all case, the last step is to store charts and recorded analysis into the project measurement repository as specified by the Data Management List.

Tailoring the Analysis Instructions:

1. *For SPI tools, remove any instructions that are not being used. This template includes procedures for more graphs than the minimal requirement.*
2. *For non-SPI tools, place the instructions for creating charts at the beginning of the instructions*
3. *For non-SPI tools, place the instructions for transferring charts to the BSR file at the end of the instructions*
4. *For non-SPI tools, the body of the instructions needs to reflect the tool being used. If the graphs are similar, edit the text that is already there, if not replace it and/or add a new procedure as appropriate.*
5. *If an Excel based BSR template is being used, change the approach to copying to reflect the use of "Move or Copy sheet" from the Edit menu.*

Software Progress

Analysis Procedure for Progress

This procedure uses two charts from different tools. The first is the Trend Chart from the SPI Point Counting Tool, which shows planned an actual progress points over time, along with a baseline for completion of the work and uncertainty bounds. The second is the Detailed Schedule chart from the SPI Schedule Tool, which shows project deliverables and major work activities for a 12 month period as a Gantt Chart.

These two types of charts are related. There are usually multiple point counting charts, as work packages on the detailed schedule, such as developing or testing a build, may each have their own point counting scheme; thus multiple copies of the Excel-based Point Counting tool are usually produced, and more than one may be active at any given time.

The analysis proceeds as follows:

- Examine Trend Data chart(s) for active task(s).
    - If progress is within the uncertainty bounds (See analysis summary in Measurement Planning Table in the Software Management Plan/Product Plan for bounds), write analysis that progress is nominal, no impact or corrective action needed. Do note if trend seems to be taking progress out of bounds in the future.
    - If progress is outside uncertainty bounds, determine the reason, assess the impact, and define corrective action if needed.
      Note: Progress can be slowed by low productivity, changing requirements, or delays waiting for needed inputs from stakeholders.
      Note: If progress is above the upper uncertainty bound, there is greater than planned progress – which may be due to high productivity, or greater than planned staffing.
      Note: Impact (delay) can be predicted by projecting when the actual progress curve will hit the baseline. Straight line projection with eyeball and ruler is generally good enough for these 2-3 month tasks.
    - Possible corrective actions include adding staff, working in other areas while waiting for inputs, delaying completion of point counting plan, or replanning work (e.g., moving capabilities to a subsequent build)
- Examine Detailed Schedule Chart
    - Look at each line to see if scheduled work has been completed. The line should be dark blue at least to the “now” line, and scheduled milestones should be made with few delays.
    - If there are significant shortfalls, the analysis should explain any work items that are either significantly ahead (dark blue well past the “now” line) or behind schedule, assess the impact, and define any corrective action needed.
      Note: The Detailed Schedule should provide evidence that a plan to make up for one shortfall by working ahead in another area is being carried out.
    - Possible corrective actions include adding staff to make deadlines, extending the schedule, or removing some of the work.

Analysis Procedure for Effort (Staffing)

This procedure uses three charts from the SPI Staffing Tool. The first is the "18 Mo Chart_Proj" chart, which has a bar graph superimposing planned and actual effort for each month past, and planned and projected effort for each future month. The second is the the "18 Mo Chart_Proj Cum" chart, which shows the cumulative effort planned, actually achieved, and projected for the same time period.

Finally, the third chart to be analyzed is the Process Effort Chart. This shows the planned and actual effort for Management, Process and Product QA, Configuration Management, Engineering, Development, and Verification & Validation. This table also provides columns to document analysis and corrective action for each process area.

The analysis proceeds as follows:

- Examine the cumulative effort chart. (18 Mo Chart_Proj Cum)
    - Compare the planned and the actual/projected curve
    - If the actual/projected line is significantly below the baseline plan line, or it is projected to fall behind in the future, determine the cause, assess the impact, and plan any necessary corrective action.
      Note: The analysis of deviations includes looking at the other two charts for this procedure.
    - Corrective action can be a change to the operating plan or a re-plan. The former makes adjustments to get back to the baseline plan, the latter actually changes the baseline plan.
      Note: Some means of getting back to baseline plan include increasing productivity through training, changing the skill mix, or introducing a process improvement. In the case where the project is understaffed, early detection allows adding staff to correct while still making budget and schedule.
      Note: Re-planning can add staff to meet higher complexity / lower productivity observed on project, extending the schedule with the current staff, or reducing the work to the amount that can be met with current staff and schedule.
- Examine the staffing chart (18 Mo Chart_Proj)
    - Compare the planned and actual/projected staff for each month.
    - If there are significant deviations, determine the cause, assess the impact, and define any needed corrective action.
      Note: The projected data is from actual assignments to personnel. If understaffing to date is causing a project to fall behind, this projection indicates that the project will continue to lose ground.
    - The corrective action for staffing issues is generally to increase or decrease staff, but reassignments, training, and process changes can be used to increase productivity.
- Examine the Process Effort chart
    - Compare the planned and actual staffing for each process area
    - If there is a deviation for any area, determine why this is, assess the impact, and plan any needed corrective action.
      Note: The impact is documented in the Analysis column
      Note: For areas with low planned staff, a few days off can show up as a high percentage variance without being an issue. Analysis should document this.
    - Corrective action includes either adding staff or shifting assignments.

Software Functionality

Analysis Procedure for Planned and Delivered Functionality by Build

This procedure uses the Functionality by Build chart from the SPI Requirements Tool. This chart shows a pair of stacked bar graphs for each CSCI. The leftmost shows the number of requirements planned for each build, the rightmost shows the number of requirements met by each build delivery. This graph does not contain any schedule information; it simply provides a means of determining whether the planned functionality is in each delivery. The analysis proceeds as follows:

- Examine the Functionality by Build chart
    - If each delivery contains the requirements planned or if graph indicates that later build has made up for a shortfall, write this result as analysis; there is no impact or corrective action needed.
    - If current delivered software is short of plan, analyze why this is the case, what the impact is, and document corrective action as needed.
      Note: possible causes include a build being more complex than expected, or delay in resolving requirements TBDs or questions.
    - Corrective actions can include writing and monitoring a risk, adding staff, or extending schedule.

Software Requirements Volatility

Analysis Procedure for Changes

This procedure uses two charts from the SPI Requirements Tool. The first, RQ Volatility, graphs the cumulative number of changes (additions, deletions or modifications) for each CSCI over time. The second, RQ TBDs, graphs the current number of TBDs for each CSCI over time. Together they are used to analyze the impact of past changes and to anticipate future changes. The analysis proceeds as follows:

- Examine the TBD chart
  - If TBDs are within expected range write analysis stating "TBDs are within range expected for project at this stage of development" with no impact or corrective action needed.
  - If there are more TBDs than expected for any CSCI, or a flat curve indicates they aren't being addressed, diagnose why, assess impact, and define corrective action if needed.
    Note: flat curves usually indicate lack of staff who are resolving TBDs
    Note: few if any TBDs should be left by CDR.
  - Corrective action for resolving TBDs can be to shift resources to address open TBDs, or to rearrange schedule to implement from more stable requirements first.
- Examine the volatility graph
  - If changes are within the expected range, write analysis stating that they are within range for this stage of development and that there is no impact and no corrective action needed.
  - If there are more changes than expected for any CSCI, diagnose why this is, assess the impact, and define corrective action if needed.
    Note: expected behavior is for the volatility graphs to flatten as time goes on
    Note: the project may see a jump in a curve or curves that reflect concentrated effort to resolve TBDs or respond to an expected requirements change. If this is anticipated by project planning, no corrective action is needed.
  - Corrective action for overly volatile software may include adding resources to resources to resolve requirements issues or rescheduling to implement or test stable portions first.
    Note: One means of addressing unstable requirements is to go through an extra cycle of requirements peer reviews to resolve remaining issues.

Analysis Procedure for Growth

This procedure uses the RQ Growth Chart from the SPI Requirements Tool. This chart graphs the current number of requirements for each CSCI over time. The analysis of this data proceeds as follows:

- Examine the graph
    - If change in total number of requirements is within expectations, write that this has occurred and that there is no impact or corrective action.
    - If number of requirements is growing too fast, determine why and write this into analysis, assess the impact, and document corrective action if needed.
      Note: Requirements growth curve should flatten as time goes on.
      Note: If new requirements are added according to plan, note that as part of statement that there is no impact.
    - Corrective actions for requirements growth may include adding staff, extending schedules, removing functionality or writing and monitoring a risk.

Software Quality

Software quality is monitored and controlled by monitoring the status and severity of problems uncovered, and by using measures to assure the effectiveness of inspections and other peer reviews. The first objective is to assure that problems are addressed in a timely manner, so analyzing the status data is of primary importance. The other data supplements this analysis.

Analysis Procedure for Status

This procedure uses two charts from the SPI Problem Report Tool. The first, Status by Date, plots the number of problem reports submitted, the number accepted as needing corrections, and the number where corrections have been completed against time. The second, Status by CSCI, provides a stacked bar graph of the current status of all problem reports for each CSCI; including intermediate states not shown on Status by Date.

The analysis of the data proceeds as follows:

- Examine the Status by Date graph
    - Look for steadily opening gap between the accepted and completed lines, or a flat completed line. If neither of these is occurring, write that the situation is within expected bounds, with no impact and no corrective action needed.
    - If the gap is steadily opening, determine the reason and assess the impact, and document corrective action if needed
      Note: Possible reasons for gap may include more errors than expected, or fewer resources than needed correct the problems. A flat or gently sloping completed line is a possible indicator of the latter situation.
      Note: Adding a vertical line to the graph at the date for the next test completion milestone will highlight the difference between a schedule impact and gaps that naturally open at the start of testing for a given build.
    - Corrective actions for lack of timely resolution of problems include adding staff, extending schedule, or writing and monitoring a risk.
- Examine the Status by CSCI chart for the current and previous period.
    - If the Status by Date data described above is as expected, look to see if any individual CSCIs are not progressing. If not, explain in analysis.
      Note: key personnel on vacation or diverted to other work is a common explanation.
      Note: in the case where the overall progress is good, corrective action is rarely needed.
    - Otherwise the gold and yellow areas representing implemented and verified corrections are increasing, there is progress that is not reflected in the Status by Date chart. If this is the case, record this in the analysis, if not determine the reason for the lack of progress, its impact, and any need for corrective action.
      Note: Part or all of analysis, impact and corrective action may be recorded once with the Status by Date chart, as long as the Status by CSCI chart is referenced.

Analysis Procedure for Severity

This procedure uses two charts from the SPI Problem Report Tool. The first, Severity by Date, plots the cumulative number of problem reports submitted by severity (Major, moderate, minor) over time. The second, Severity by CSCI, provides a stacked bar graph of that indicates the number of defects of each severity for each CSCI. Like the Status by CSCI graph, this represents data at the current time, rather than a trend.

The analysis of severity data proceeds as follows:

- Examine the Severity by Date chart
    - Look for jumps in the number of major errors, or to see if the proportion of major errors seems high. In either of these cases, determine the cause, assess the impact and plan any corrective action needed.
- Examine the Severity by CSCI chart
    - Look to see if major errors are clustered in a particular CSCI or small number of CSCIs. If they are determine the cause, assess the impact and plan any corrective action needed.
      Note: typical causes include lack of experienced personnel or a particularly complex set of requirements.
- If status data shows that errors are being resolved in a timely manner, there is no impact, and corrective action is optional. One might still consider looking at changes that might prevent major errors on future builds; e.g., more stringent code inspections, or assigning more senior person to help with error prone CSCIs.

Analysis Procedure for Inspection Data

This procedure uses the Status Data Worksheet from the Inspection Metrics Tool. This information is used to determine if a particular work product is error prone and if the inspection process is being carried out effectively.

The analysis proceeds as follows:

- Examine and analyze the following data items:
  - *Planned and actual number of meetings* (found in upper left corner of worksheet). If all the planned meetings are not held, the reasons are explained as part of the analysis.
  - *Meeting Length* If meetings are commonly lasting more than two hours, note it in your analysis, note the impact of reduced effectiveness in removing defects before test, and plan corrective action.
  - *Number of defects found* Look for inspections that had an exceptionally high number of defects or an exceptionally low number of defects and explain the cause of this in the analysis
    Note: This is not necessarily bad news; it can represent a job well done by an inspection team or an author, respectively.
  - *Number of defects deferred* If defects are persistently being deferred, this indicates that authors are persistently missing information they need to complete their work. Analyze why this is so, and in the case where these deferred actions are not being resolved take corrective action to either resolve them more quickly or prevent the lack of information from occurring in the first place.
  - *Action Items* (if inspection actions are tracked in this tool) Look for inspections where action items are remaining open for more than a month; explain the cause as part of analysis
    Note: The most common delay is waiting for information from relevant stakeholders outside the team.
- Document the analysis, impact, and corrective action in the box below the table
  Note: For presentation of data, hide rows from previous months, except if action items are being tracked in the tool and actions from a given inspection remain open.
  Note: Most common impact of problems with inspection process is increased risk of problems during testing and of delivering software to the customer while still containing defects.